

# PATENT ABSTRACTS OF JAPAN

(11) Publication number : 2003-347927

(43) Date of publication of application : 05.12.2003

(51) Int.Cl. H03K 19/173

(21) Application number : 2002-152847 (71) Applicant : NIPPON TELEGR & TELEPH

CORP <NTT>

(22) Date of filing : 27.05.2002

(72) Inventor : TSUBOI HIDEYUKI

SHIOZAWA TSUNEMICHI

INAMORI MINORU

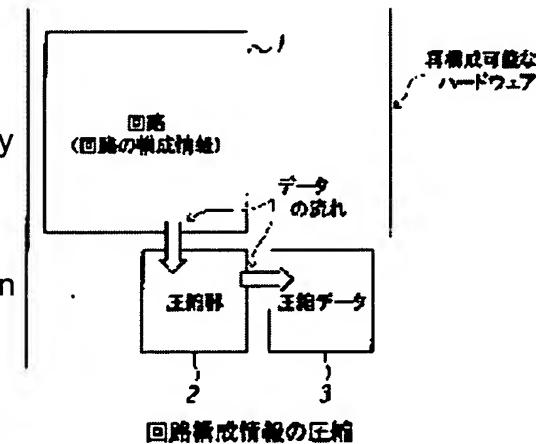
NAKANE YOSHIKI

(54) DATA PROCESSING CIRCUIT FOR RECONFIGURABLE HARDWARE AND METHOD THEREFOR

(57) Abstract:

PROBLEM TO BE SOLVED: To provide a data processing circuit capable of reducing the capacity of input and output data, performing high speed input and output operations, freely realizing a compressor and an expander or the like at a required position and effectively utilizing hardware resources.

SOLUTION: The configuration information 1 of the data processing circuit is supplied to a compressor 2 along an arrow (data flow) shown in Figure. The capacity of data compressed via the compressor 2 is decreased. The compressed data whose capacity is made compact are stored as compressed data 3 according to a data flow indicated in the succeeding arrow. When the compressor 2 is replaced with an expander, the data flow follows the reversed arrows, and the expander restores the internal compressed data 3 into the configuration information 1 of the data processing circuit for further processes.



## LEGAL STATUS

[Date of request for examination] 05.07.2004

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開2003-347927

(P2003-347927A)

(43)公開日 平成15年12月5日 (2003.12.5)

(51)Int.Cl.<sup>7</sup>

H 03 K 19/173

識別記号

1 0 1

F I

H 03 K 19/173

テーマコード(参考)

1 0 1 5 J 0 4 2

審査請求 未請求 請求項の数13 O.L (全 17 頁)

(21)出願番号

特願2002-152847(P2002-152847)

(22)出願日

平成14年5月27日 (2002.5.27)

(71)出願人 000004226

日本電信電話株式会社

東京都千代田区大手町二丁目3番1号

(72)発明者 坪井 秀幸

東京都千代田区大手町二丁目3番1号

本電信電話株式会社内

(72)発明者 塩澤 恒道

東京都千代田区大手町二丁目3番1号

本電信電話株式会社内

(74)代理人 100077274

弁理士 磯村 雅俊 (外1名)

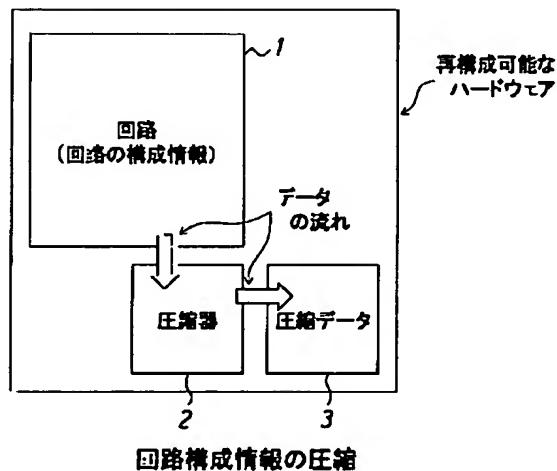
最終頁に続く

(54)【発明の名称】 再構成可能なハードウェアにおけるデータ処理回路およびその方法

(57)【要約】

【課題】入出力データの容量抑制が可能であり、高速な入出力動作を行うことが可能であり、圧縮・伸長器などが必要な位置に自在に実現でき、ハードウェア・リソースを有効に活用できるデータ処理回路を提供する。

【解決手段】回路の構成情報1を矢印(データの流れ)に従って圧縮器2へ入れる。この圧縮器2を介して圧縮したデータの量は少ない容量になる。コンパクトにできた圧縮データは、その次の矢印に示すデータの流れに従って圧縮データ3をして格納される。一方、圧縮器2の代りに伸張器を置き換えると、矢印を逆に向けた流れとなり、内部の圧縮データ3が伸張器により回路の構成情報1に復元され、さらに次の処理を行うことができる。



【特許請求の範囲】

【請求項1】 再構成可能なハードウェアにより様々なデータの処理を行うデータ処理回路において、上記様々なデータの処理に介在する圧縮器あるいは符号器を、上記データ処理回路内に内包して具備することを特徴とする再構成可能なハードウェアにおけるデータ処理回路。

【請求項2】 再構成可能なハードウェアにより様々なデータの処理を行うデータ処理回路において、上記様々なデータの処理に介在する伸張器、解凍器または復号器を、上記データ処理回路内に内包して具備することを特徴とする再構成可能なハードウェアにおけるデータ処理回路。

【請求項3】 再構成可能なハードウェアにより様々なデータの処理を行うデータ処理回路において、上記様々なデータの処理には複数のデータの処理が組み合わさって実行され、上記各データの処理の少なくとも1箇所に介在する圧縮器あるいは符号器と、かつまた、上記各データの処理の少なくとも別の1箇所に介在する伸張器あるいは解凍器または復号器とを、上記データ処理回路内に内包して具備することを特徴とする再構成可能なハードウェアにおけるデータ処理回路。

【請求項4】 再構成可能なハードウェアにより様々なデータの処理を行うデータ処理方法において、上記再構成可能なハードウェア内の回路の構成情報を多量かつ高速転送により該ハードウェア内の圧縮器に入力し、入力された情報を該圧縮器により圧縮データに変換し、該変換された圧縮データを該圧縮器から少量かつ低速転送により外部に出力して、退避させることを特徴とするデータ処理方法。

【請求項5】 再構成可能なハードウェアにより様々なデータの処理を行うデータ処理方法において、外部に退避した圧縮データ、またはソフトウェアにより作成された同一データを、上記再構成可能なハードウェア内の伸張器に入力し、入力された圧縮データを該伸張器により伸張し、回路の構成情報に変換することを特徴とするデータ処理方法。

【請求項6】 再構成可能なハードウェアにより様々なデータの処理を行うデータ処理方法において、上記再構成可能なハードウェア内の回路の構成情報を該ハードウェア内の圧縮器に入力し、入力された情報を該圧縮器により圧縮データに変換し、該変換された圧縮データを該ハードウェア内のメモリに格納することを特徴とするデータ処理方法。

【請求項7】 再構成可能なハードウェアにより様々なデータの処理を行うデータ処理方法において、上記再構成可能なハードウェア内のメモリに格納された圧縮データを、該ハードウェア内の伸張器により伸張し、回路の構成情報を復元することを特徴とするデータ処理方法。

【請求項8】 再構成可能なハードウェアにより様々なデータ処理を行うデータ処理方法において、上記再構成可能なハードウェア内の1つ以上のメモリに格納された複数の圧縮データを、1つの伸張器に任意の順番で転送し、該伸張器により順次回路の構成情報に復元することを特徴とするデータ処理方法。

【請求項9】 再構成可能なハードウェアにより様々なデータ処理を行うデータ処理方法において、外部に退避した圧縮データまたは外部からの回路構成情報を、上記再構成可能なハードウェアの回路またはメモリに入力し、該回路の回路構成情報を該ハードウェア内の圧縮器で圧縮データに変換し、一方、該メモリ内の圧縮データを該ハードウェア内の伸張器で伸張して回路構成情報を復元することを特徴とするデータ処理方法。

【請求項10】 請求項1、2または3に記載の再構成可能なハードウェアによるデータ処理回路、または請求項4に記載のデータ処理方法において、前記圧縮器、符号器、伸張器、復号器、または解凍器は、いずれもプログラムにより実現され、ハードウェア内の任意の位置に移され、あるいは消去されることが可能であることを特徴とする再構成可能なハードウェアによるデータ処理回路。

【請求項11】 請求項5～9のいずれかに記載のデータ処理方法において、前記圧縮器、伸張器またはメモリは、いずれもプログラムにより実現され、ハードウェア内の任意の位置に移され、あるいは消去されることが可能であり、該メモリは専用の格納エリアを設けることなく、任意の位置に圧縮データまたはその他のデータを格納することが可能であることを特徴とするデータ処理方法。

【請求項12】 再構成可能なハードウェアにより様々なデータの処理を行うデータ処理回路において、連長圧縮アルゴリズムをPCAに実現した圧縮器を具備し、該圧縮器は、組込部と可変部間のデータ幅に従って、入力値を保持する部分、連続して同じ入力をカウントする部分とを持ち、異なる値の入力で、保持した値とカウントした値を出力する間の出力状態保持部を有することを特徴とする再構成可能なハードウェアによるデータ処理回路。

【請求項13】 再構成可能なハードウェアにより様々なデータの処理を行うデータ処理回路において、連長圧縮アルゴリズムをPCAに実現した伸張器を具備し、

該伸張器は、組込部と可変部間のデータ幅に従って、連続する2回の入力で最初の値を保持する部分、次の値をセットするカウンタとを持ち、それぞれ1回の入力毎に入力状態を切り替える入力状態保持部を有し、該カウンタがカウントした値を減少させながら、該入力状態保持部は保持した値を出力することを特徴とする再

構成可能なハードウェアによるデータ処理回路。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、ハードウェアにより様々なデータ処理を行う回路に関し、特に出入力データの容量抑制、高速な入出力動作をするデータ処理回路の実現が可能な再構成可能なハードウェアにおけるデータ処理回路に関する。なお、回路の構成情報を処理するデータにする場合には、再構成可能なハードウェア（RH：Reconfigurable Hardware）というタイプの集積回路に適用することもできる。RHには、FPGA（Field Programmable Gate Array）、CPLD（Complex Programmable Logic Device）、PCA（Plastic Cell Architecture）などがある。また、本技術を一般的なデータの処理に対して用いるようにすれば、汎用的な／信号処理プロセッサ（MPU：Micro Processing Unit／DSP：Digital Signal Processor）をはじめ、特定の機能を持つ集積回路（ASIC：Application Specific Integrated Circuit）にも適用できる。

【0002】

【従来の技術】まず、従来のRHの利用例について述べる。図7は、RHにユーザーが回路を再構成する従来例1を示した図である。図7を従来例1としてRHの回路プログラムについて述べる。まず、ユーザーは外部から最初の構成情報D'a'を再構成可能なハードウェア（以下、RH）に入力する。構成情報D'a'はデータの流れ

（1）に従い、1つ目の回路A'9が実現される。ある間に、この実現された回路A'9が動作して信号の処理が行われる。その後、別の動作の信号処理が必要になると、次にユーザーは構成情報D'b'をRHに入力し、データの流れ（2）に従い、回路B'9が実現される。さらに別の機能が必要になれば、上記のような手順を繰り返したり、元の回路A'に戻すにも同様にRHの外部から構成情報D'a'を入力したりする必要がある。

【0003】図8は、従来のRHの利用例（複製機能）である従来例2を示した図である。図8では、RHの一部に既に再構成された回路C'10などがあり、これを別の場所に同じ回路D'11を実現する例である。RHでの回路複製機能について述べる。回路C'10から回路の構成情報が図中の矢印に示すデータの流れのように回路D'の位置へ運ばれ、回路D'11が実現されることを示している。この従来例2のRHの一例として、PCAの上で回路を複製する機能に関しては、特開平11-167556号公報（特願平9-332737号）“論理回路の動的な構成方法”にも述べられている。

【0004】また、下記に挙げる（イ）（ロ）の参考文献にも取り上げられており、さらに（ハ）の参考文献（研究会資料）においては詳細な説明がされている。

（イ）中田 広、伊藤 秀之、小西 隆介：“完全非同期回路によるPCAハードウェアの設計・評価

,” NTT R&D, Vol. 49, No. 9, pp. 518-526, Sep. 2000.

（ロ）塩澤 恒道、永見 康一、Norbert Imlig、小西 隆介：“PCA アプリケーションと設計環境,” NTT R&D, Vol. 49, No. 9, pp. 527-536, Sep. 2000.

（ハ）小西 隆介、伊藤 秀之、中田 広、塩澤 恒道、稻森稔、名古屋 彰：“非同期式動的再構成可能 LSI による自己複製回路,”電子情報通信学会技術研究報告, VLD2000-79, ICD2000-136, FTS 2000-44, pp.59-64, Nov. 2000.

【0005】なお、回路構成情報を圧縮する機構を持つFPGAと圧縮方法としては、例えば、特開2001-28536号公報“プログラマブル論理回路装置、情報処理システム、プログラマブル論理回路装置への回路の再構成方法、プログラマブル論理回路装置用の回路情報圧縮方法”（以下、従来例3）に記載されている。上記公報には、プログラマブル論理回路装置に、回路情報記憶手段と、この記憶手段に記憶された回路情報を用いて指定された回路の回路情報を生成する回路情報編集手段とを設け、回路情報記憶手段には、複数個の回路の回路情報を圧縮した状態で格納するものである。

【0006】

【発明が解決しようとする課題】従来の技術では、必要な時には態々RHの外部から構成情報を入力する必要があった（従来例1）。あるいは、従来例2に示したようにRHの内部に回路C'10を保持しておき、その回路C'10を複製するなどの処理をしなければならなかつた。外部からの構成情報などのデータを入力する場合（あるいは、出力する場合も同じである）においては、通常、集積回路等の内部処理速度に比較すると遅いというような不利な点があるとされている。また一方で、ある回路10を内部に保持した従来例2は、処理速度の面では有利であるものの、回路C'の保持に必要となる余分な面積的な容量、領域的に十分なオーバヘッドを見込んでおくことが必要となる。

【0007】上記従来例3では、論理回路を実現するプログラム論理回路部と回路情報入力制御部とが別々の機構であり、また回路情報入力制御部に圧縮する機構が含まれている。さらに、回路記憶部という専用の格納エリアも具備している。従って、固定的なそれぞれ個別部分には、これを利用する際に制限を受けるとともに、柔軟性に欠ける、という問題点がある。また、別の用途に利用することもできないという問題点もある。

【0008】そこで、本発明の目的は、これら従来の課題を解決し、入出力データの容量抑制が可能であり、高速な入出力動作を行うことが可能であり、圧縮・伸長器などが必要な位置に自在に実現でき、ハードウェア・リソースを有効に活用できるデータ処理回路を提供することにある。

【0009】

【課題を解決するための手段】上記目的を達成するため、本願の第1の発明は、R Hの内部に圧縮（符号）器を持たせるものである。内部の圧縮（符号）器により、R H上に実現する回路の構成情報を圧縮（符号）器でコンパクトにして、その圧縮（符号）データを保持し、また取り扱う。図1は、圧縮器2を用いて回路の構成情報1をコンパクトにした圧縮データ3を格納する状態を示している。すなわち、図1のデータ処理回路は、様々なデータの処理に介在する圧縮器2あるいは符号器を、上記データ処理回路内に内包して具備するものである。回路1は、レジスタ、論理ゲート、スイッチ等で構成された回路であって、任意の論理回路が想定される。（請求項1、4、6、12に対応）

【0010】回路の構成情報1を、最初の矢印に示すデータの流れに従い圧縮器2へ入れる。この圧縮器2を介して圧縮したデータの量は少ない容量になる。このコンパクトにできた圧縮データは、その次の矢印に示すデータの流れに従い圧縮データ3として格納される。これにより、R Hの上では、回路よりも面積的に小さな領域に圧縮データ3が収められる。このために、様々な回路を次々に動的に再構成しながらデータ処理を行う用途において、R Hの限られたリソース全体をより有効に活用することが可能になる。圧縮器2から圧縮データ3へのデータの流れは、回路1から圧縮器2へのデータの流れよりも低速である。この速度に差がある性質を効果的に活用すれば、外部に圧縮データ3を取り出す場合には、内部の処理速度を高速に保持して出力（インターフェース）処理速度を抑制できる。

【0011】第2の発明は、R Hの内部に伸張（解凍／復号）器4を持たせるものである。この伸張（解凍／復号）器4により、上記の圧縮（符号）器2で処理されたり、外部から入力されたりした圧縮（符号）データ3を処理する回路を実現する。図2においては、回路構成情報の伸張は、圧縮データ3から伸張器4を介して回路1を実現する状況を示している。

【0012】図1で格納された圧縮データ3を、矢印のデータの流れに従い伸張器4へ入力する。この伸張器4を介して圧縮データは回路の構成情報1に戻り、この構成情報は次の矢印で示すデータの流れに従い、回路1が実現（復元）される。この様に、回路が占める領域よりも狭い領域に格納された圧縮データ3と伸張器4を用いることで、必要な時に回路1を実現（復元）させることができ、R Hの面積的に限られた領域をこれまでよりも有効に使用することができる。また、圧縮とは逆の伸張の過程では、圧縮データ3から伸張器4までのデータの流れは、その伸張器4から回路の構成情報を送る流れにおける速度に比べ相対的に低速となる。従って、外部から圧縮データをR Hに入力して、R Hの内部の伸張器4を介して回路1を実現する場合、外部入力の速度が低速

な状況でも、回路1は高速に実現できる、というメリットが生まれる。（請求項2、5、7、8、13に対応）

【0013】本願の第3の発明は、圧縮（符号）と伸張（解凍／復号）と、さらに必要ならばこれらに関連する圧縮（符号）データとを、R Hに内包して具備する。この第3の発明により、R Hに限られたリソース内で、必要に応じて自在かつ柔軟に次々と異なる回路をR Hに実現できる。この手法による回路実現の形態は、従来のR Hに比べて外部とのやり取りに掛かる速度の課題を軽減したり、解消したりすることができると同時に、R Hの領域的な面積の利用効率も高くすることができ、R Hシステム全体のサイズを小さく抑える効果も生じる。（請求項3、9、10、11に対応）なお、第1、第2、第3の発明においては、圧縮器や伸張器などの他の回路 자체をプログラムして実現するものであり、従って、圧縮や伸張やその他の回路を任意の位置に移したり、不要な時には消去して、別の機能を上書きするなど、自在に扱うことが可能である。

【0014】

【発明の実施の形態】以下、本発明の実施形態ならびに実施例を、図面により詳細に説明する。

（第1の実施形態）図3は、本発明の第1の実施形態を示す圧縮の利用例（外部退避）の説明図である。図3では、まず、R H上で、回路A 1からデータの流れ（1）の矢印に従い圧縮器2へ、回路の構成情報が入力される。次にその回路A 1の構成情報は圧縮器2によって圧縮データD aに変換される。このとき、圧縮データはデータの流れ（2）の矢印に従い、圧縮器2からR Hの外部へ圧縮データD aが出力される。図3において、2つのデータの流れ（1）と（2）のデータ量およびその速度を比較すると、データの流れ（1）のデータ量は回路A 1の構成情報そのままであるため多量かつ高速である。一方、同じ回路A 1のデータの流れ（1）に対して、データの流れ（2）のデータ量は回路A 1の構成情報が圧縮され、その圧縮データD aは少量かつ低速で扱われる。このことから、外部出力インターフェースが低速で圧縮データD aを扱う時も、内部の圧縮処理は高速に動作しており、結果的に回路A 1の構成情報を高速に外部退避させることができる。

【0015】（第2の実施形態）図4は、本発明の第2の実施形態を示す伸張の利用例（プログラム）の説明図である。第1の実施形態の圧縮器2の代わりに伸張器4をR Hの内部に持つ第2の実施形態は、図4に示すように伸張の利用例（プログラム）を示している。第1の実施形態において、外部に退避した圧縮データD a、またはソフトウェアにより意図的に同じデータを作成しても良いが、そのデータD a（回路A 1の構成情報に変わる）をR Hに入力する。R Hの内部の伸張器4へ、圧縮データD aが矢印で示すデータの流れ（1）に従って入力する。この伸張器4では、圧縮データD aが伸張さ

れ、回路A 1の構成情報に変換される。変換された構成情報は、その次の（右側の）矢印で示すデータの流れ（2）に従い回路A 1の位置で実現される。

【0016】ここまで述べた中の2つのデータの流れ（1）と（2）の関係は、前記第1の実施例とは逆に、データの流れ（1）に比べて、データの流れ（2）ではデータ量は多く、かつ高速である。従って、圧縮データD aを入力する所のインターフェースが低速な処理しか扱えなくとも、RH内部に伸張器4を持つならば、この伸張器4を介することで、回路A 1を高速に実現できる。

【0017】（第3の実施形態）図5は、本発明の第3の実施形態を示す異なる回路伸張の利用例の説明図である。第2の実施形態のように、伸張器4をRHの内部に持つ場合でも、これにさらに加えて複数の圧縮データ7, 8を内部に持つ別のケースとなる第3の実施形態を図5に示している。図5では、1つの伸張器4と、圧縮データD a 7と圧縮データD a 8の2つの異なる圧縮データがRH内にある。これは、第2の実施例の外部に位置した圧縮データD aがRH内にあると考えればよい。2つの異なる圧縮データ7, 8があるが、順番に圧縮データD aは伸張器4を介して、データの流れ（a）に従い、回路A 5が実現（復元）される。また、圧縮データD bも同じ伸張器4を介し、データの流れ（b）に従い、回路B 5が実現される。

【0018】ここで回路A 5と回路B 5は、RHの同じ位置の領域（全く同じ領域である必然性ではなく、回路AとBで占める領域の一部分が一致しても良い）を使用できる。つまり、実現する時間を異にすることで、RHの同じリソースを回路AとBは共有し、有效地に利用できる。また、回路Aを復元する処理速度は、これまで説明したように、外部とのインターフェースを一切介さないため、第2の実施例の圧縮データD aをRHの外部から入力する場合よりもさらに高速にできる。

【0019】さらに、回路Aに外部入力信号、外部出力信号、内部入出力信号の全部またはどれか部分的にでもあるとき、これらの経路を当然のことであるが回路Aは使用していることになる。一方で、回路Bが圧縮データD bから実現されたときに、回路Aで使われていた外部入力信号、外部出力信号、内部入出力信号の全部またはどれか部分的でも経路を全部または一部をそのまま使える。逆に、回路Bも、外部入力信号、外部出力信号、内部入出力信号の全部またはどれか部分的である。これらの経路が存在するので、圧縮データD aから実現された回路Aが、残存する経路を全部または一部をそのまま使うことができる。

【0020】つまり、回路Aと回路Bは、互いにRHの同じ位置の領域を共有しているだけでなく、外部入力信号、外部出力信号、内部入出力信号の経路も全部あるいは一部を共有することができる。従って、回路Aと回路Bの外部入力信号、外部出力信号、内部入出力信号の経

路を別にする必要もないし、予め回路の設計段階からそれらの経路も検討することはない。むしろ、この第3の実施形態において、回路Aと一致する外部入力信号、外部出力信号、内部入出力信号の経路を、回路Bが積極的に活用できると考えられる。

【0021】図5では、2つの圧縮データを示した。すなわち、圧縮データD aとD b（回路A, B）は説明の都合により用いる順番に触れた。利用の必要に応じて、この順番を逆にすることも可能である。また、図5で圧縮データの数“2”も、さらに多数（複数）の圧縮データを持つ場合も、RHのリソースの許す範囲内で考えられる。この場合には、必要に応じてそれらの圧縮データ（復元する回路）を選択し利用することができる。

【0022】（第4の実施形態）図6は、本発明の第4の実施形態を示す回路圧縮・伸張の利用例の説明図である。図6では、第1の実施形態の圧縮器2と、第2の実施形態の伸張器4と、さらに第3の実施形態の圧縮データ7, 8とを、全てRHの内部に持つ場合の実施形態である回路圧縮・伸張の利用例を示している。図6には、大きく3つのデータの流れ（1）, (2), (3)が関与している。説明が簡単な順に、図1と同様で、圧縮で説明したデータの流れ（1）は、回路Aが圧縮器2を介して圧縮データD a 7へ格納する過程を示す。また、第3の実施例である図5と同じ過程では、圧縮データD a 7からデータの流れ（3）に従い、順に伸張器4によりデータが伸張され、回路A 5になる。

【0023】さらに、上記従来例1の図7で説明した過程のように、外部の回路構成情報D cをデータの流れ（2）に従いRHに入力して回路C 5を実現させる。ちなみに、上記データの流れ（1）は、データの流れ（3）の過程とは逆になる。この3種類の過程を持つことにより、回路A 5を一時的にRH内に圧縮データとして格納し、回路C 5を再構成して実現し、割り込ませるような形態の処理を実施し、再び回路A 5を復元させることができ、限られたRHのリソースで高速に実行できる。

【0024】加えて、回路A 5に外部入力信号、外部出力信号、内部入出力信号の全部またはどれか部分的にでもあるとき、これらの経路を当然のことながら回路A 5は使用することになる。一方で、外部の圧縮データD cをRHに入力して回路C 5が実現されたときに、回路A 5で使われていた外部入力信号、外部出力信号、内部入出力信号の全部あるいはどれか部分的でも、回路C 5の経路としてそのまま使用できる。更に、回路C 5を実現した領域に、回路A 5を再度実現する際にも、各入出力信号の全部またはどれか部分的にあり、これらの経路があるので、圧縮データD a 7から実現された回路A 5が、それらの経路の全部または一部をそのまま使用できる。

【0025】すなわち、回路A 5と回路C 5は互いにRHの同じ位置の領域を共有し、さらに、外部入力信号、

外部出力信号、内部入出力信号の経路も全部あるいは一部を共有できる。従って、回路A5と回路C5の外部入力信号、外部出力信号、内部入出力信号の経路を態々別に設ける必要もないし、予め回路の設計段階から別の経路設計も不要である。この第4の実施例では、積極的に回路Aと一致できる外部入力信号、外部出力信号、内部入出力信号の経路を回路Cが活用できる。

【0026】また、データの流れ(2)による再構成で実現した回路C5をデータの流れ(1)と(3)に従い、圧縮データD<sub>a</sub>7／回路A5に置換えて、圧縮データD<sub>c</sub>／回路Cというように操作・制御することも自在に実施できる。このようにすれば、次々と別の回路を一時的にRH内部に格納しておき、必要な時には高速に回路を復元させることも可能となる。さらに、データの流れ(2)に伸張器(別に設けなくとも内在している同じ伸張器を共有し利用することも可能である)4を介在させれば、第2の実施形態の図4で述べたように、外部から圧縮データによる回路Cの実現を高速にさせることもできる。

【0027】あるいは、図6で、圧縮データD<sub>a</sub>7の所に、外部から直に圧縮データD<sub>c</sub>をデータの流れ(4)に従い、書き込むことで、回路A5が動作時でも、この書き込みは同時並行的に実現できる。このように書き込みをした場合には、その後にデータの流れ(1)と(3)に従い、圧縮データD<sub>c</sub>から回路C5を再構成することは、RH内部のみの処理に当たることから、より高速な実現が可能となる。

【0028】(PCAでのシミュレーション)次に、前述の第1～4の実施形態で用いられる圧縮器、伸張器に関し、RHとしてPCAの場合について、その圧縮器、伸張器の単純な原理・構成の一例を挙げ、シミュレーションでそれらの基本的な動作を説明する。なお、PCAについては、下記文献で概要から詳細まで述べられている。(二)名古屋 彰、小栗 清:「プラスティックセラーアキテキチャ(PCA)技術の概要」NTT R&D, Vol.4 9, No.9, pp.513-517, Sep. 2000. (ホ)小栗 清:「布線論理による新しい汎用情報処理アーキテクチャPCA(1,2,完)」bit, Vol.32, No.1, pp.27-35, Jan., No.3, pp.54-62, Mar., No.7, pp.51-59, Jul. 2000.

【0029】そこで、PCAは、任意の回路や情報を格納するメモリなどオブジェクトを実現する可変部(P<sub>P</sub>: Plastic Part)と、複数のオブジェクト間や外部とオブジェクト間の情報を通信する組込部(B<sub>P</sub>: Built-in Part)がある。本発明では、圧縮に関する元回路や圧縮器・伸張器はPPでオブジェクトとして実現される。以下、PCA上の圧縮・伸張に関し、簡単にその実現と動作を説明する。

【0030】図17は、PCAに実現した圧縮器の画面図であり、図18は、PCAに実現した伸張器の画面図である。図17および図18により、データ処理に用い

るため、連長圧縮アルゴリズムをPCAに実現した圧縮器および伸張器について、その構成を説明する。まず、圧縮器は、組込部(B<sub>P</sub>)と可変部(P<sub>P</sub>)間のデータ幅に従って、入力値を保持する部分、連続して同じ入力をカウントする部分とを持ち、異なる値の入力で、保持した値とカウントした値を出力する間の出力状態保持部を有することを特徴としている。また、伸張器は、組込部と可変部間のデータ幅に従って、連続する2回の入力で最初の値を保持する部分、次の値をセットするカウンタ、とを持ち、それぞれ1回の入力毎に入力状態を切り替える入力状態保持部を有し、そのカウントした値を減少させながら、保持した値を出力することを特徴としている。

【0031】すなわち、圧縮器および伸張器は、とともに、まずPCAのBPとPP間でやり取りするデータの幅に従い、連長圧縮における「コード」および「連長数」を設定している。本願実施例に示す圧縮器および伸張器は、前述した参考文献、特に(二)、(イ)で取り上げているように、試作したPCAのプロトタイプLSIであるPCA-1で実現した。図17のPCAに実現した圧縮器および図18のPCAに実現した伸張器について、さらに説明する。このPCA-1で圧縮器および伸張器を実現した場合、そのBP-PP間のデータ幅は入力・出力ともに4bitである。このような扱うデータの条件に従い、PCAに実現した連長圧縮における「コード」は'0'～'FF'(16進数表示)，連長数は最大'31'とする。また、「コード」および「連長数」の設定に加え、一方の圧縮器については、入力値が時間的に直前の入力値と一致する時は、出力をせず内部カウンタを'+1'と増加させる。直前と異なる値が入力された時は、出力状態保持部が出力状態に切り替わる。そして、直前まで入力されていた値(コード)と、内部カウンタの値(連長数)を続けて出力する。その後は、内部カウンタをリセットし、また出力状態保持部が入力状態に戻し、最初から同じ手続きで動作を繰り返していく構成としている。

【0032】この圧縮器の内部構成で、入力値を保持する部分は、組込部と可変部間のデータ幅に合わせた4bitであり、「0'～'FF'(16進数表示)''を保持できる。また、内部カウンタは、同じ4bitで'0'～'31'をカウントできる。他方、伸張器については、入力を連続して2回取り入れる。最初の値をバッファに保持し、入力状態保持部が最初から次の状態に切り替わる。次の値は、カウンタにセットする。この状態のまま、このカウンタを'-1'と減少させながら、バッファの値を繰り返し出力する。カウンタの値が'0'となったならば、入力状態保持部を最初の状態に戻し、最初から同じ手順を繰り返して動作する構成としている。この伸張器の内部構成は、圧縮器と同じように、最初の入力値を保持する部分が、4bitバッファ

で‘0’～‘FF’（16進数表示）”を保持できる。また、次の値をセットする4bitカウンタも‘0’～‘31’をカウントできる。なお、PCAのPPに設計された回路の論理素子などの表記に関する説明（例えば、分岐、andゲート、orゲート、xorゲート、交差、インバータ、norゲート、セレクタなど）は、例えば、第6回パルテノン研究会発表（2）‘PCAの実現を支援するソフトウェア’小西隆介他著を参照されたい。

【0033】（第1、第4の実施形態に関する実施例1）図9は、本発明の実施例1を示す圧縮器と動作の状況の画面およびその説明図である。図9には、上記PPの状況を示しており、PCAに実現される様々な回路の動作をシミュレーションする画面で、ある時間が経過した所で回路が圧縮動作している状況である。この画面の左上隅に1ブロック（PCAでは、このブロックをPC Ace11と名付けている）分の領域を占める回路X1がある。その回路1の下方向に接するように4ブロック（ $2 \times 2$  PC Ace11s）分の領域に圧縮器2を配置している。回路X1の構成情報を圧縮したデータ15（単に、圧縮データと呼ぶこととする）は、その圧縮器2の右隣にメモリとして置き換えられている。なお、説明を分かり易くするため、シミュレータの画面上にまだ回路などオブジェクトが存在していない領域に、位置の目安となる点線で四角の枠を幾つか表示している。

【0034】さて、ここに例示した回路X1は、簡単な圧縮動作の説明のために、ここでは僅かに1 PC Ace11分の領域を占める回路を取り扱っている。しかし、実際には、多数のブロック分（ $M \times N$  PC Ace11s, M, N: 任意の数）の領域を占める回路を圧縮することが有効である。また、ここで挙げている圧縮器のアルゴリズムは、原理・構成が単純な連長圧縮法（Run Length Encoding）である。連長圧縮法は、連続したデータを‘コード’と‘連続数’に変換する。この連続圧縮法に多少の工夫を加えて、様々な処理に適用する事例が多い。この連長圧縮法を応用して画像処理へ適用する事例については、例えば、特開2000-125111 画像処理装置およびその方法がある。

【0035】特に、この連長圧縮法は、ディジタル動画像圧縮の中でも可変長符号化（Variable-Length Coding）として最も多くの適用検討がなされている。（特許庁ホームページ、技術分野別特許マップ、電気14ディジタル動画像圧縮技術第1章特許からみた技術開発の動向1.3.5 符号割り当て（1）可変長符号化 <http://www.jpo.go.jp/ryutu/map/dekki14/1/1-3-5-1.htm>）ここで説明に用いる連長圧縮法の圧縮器では、4bit幅のデータ列を扱う。従って、前記実施例のような適用には、さらに幅の広いデータ列を扱うようにしたり、あるいはアルゴリズムもより圧縮効率の高い圧縮器を使うようにしたりすることも考えられ

る。

【0036】ちなみに、nbit幅のデータ列で連長圧縮法を用いた場合の、論理的な最大圧縮率を述べる。同じデータ（コード）が連続して $2^n$ 回続き入力されると、出力が‘コード’のnbitと‘連続長’のnbitとなる。この入力と出力のサイズは、それぞれ $2^n \times nbit$ ,  $nbit + nbit$ であるから、最大圧縮率は、 $1/2^{n-1}$  ( $= 2^n / 2^n \cdot n$ ) となる。つまり、この説明に用いた圧縮器では4bit幅までしか扱えないが、データの幅を8bitへ単に拡張することにより、この圧縮率が $1/128$ にもよくすることができる。従って、連長圧縮法においては入力されるデータ列の各データ幅を少しでも広くすると圧縮率を急激に高く改善できる可能性がある。さらに、圧縮のアルゴリズムを連長圧縮法から別の圧縮法（例えば、一般的なものには、ハフマン符号、算術符号化、Lempel-Ziv（LZ）などがある。）に代えれば、一般的に連長圧縮法に比べて圧縮率は高いと言われている。

【0037】図9は、回路の詳細な画像を示しているが、これに対応する大まかな可変部（PP）のようすを示したものが、図10である。

【0038】図10は、圧縮動作の概略を示す画面および説明図である。図10は、図9と同様に、左上隅から順に回路1、この下に接して圧縮器2、さらに右隣に圧縮データ15となっている。圧縮データ15は、回路X1（この領域は1ブロック分、すなわち、1 PC Ace11）の構成情報を圧縮したものであり、ここでは回路X1と圧縮器2の場合には、シミュレータ画面のほぼ中央に点線で示す小さな四角よりも小さな領域に収まっている。以上、図9と図10では、回路の構成情報を圧縮する動作を途中の状況で、PCAのPPを説明した。

【0039】図11は、圧縮データの流れを示す画面および説明図である。この図11はPCAのBPを示しており、同じ圧縮動作を行っているシミュレータの画面である。PPの各ブロックに対応するBPでの同じ場所を分かるように、シミュレータ画面上に四角の枠を描いている。実線の四角の枠で左上隅は、回路X1がPPに実現されている対応したBPの位置を示している。その下側に接する実線の大きな四角の枠は、PPにある圧縮器に位置および領域が対応しているBPである。この右隣に点線の四角に圧縮データが書き込まれるPPに対応するBPを示す。なお、以降で、伸張の説明に関するオブジェクトがPPに作られる領域に対応するBPにも幾つかの点線の四角の枠を描いてある。

【0040】図11に示すPCAのBPでは、圧縮データの流れ16を確認することができる。回路X1の部分では、自身の回路の構成情報を可変部（PP）から読み出され、下側の圧縮器2のあるBPへの経路に沿って回路の構成情報が一連のデータ列となって流れている。こ

の画面では、丁度、回路X1と圧縮器2の間となるBPの経路上に一連の回路構成情報の一端となるデータ'8'（16進数で表示されている）が示されている。そして、圧縮器2のあるPPへ一旦入り、圧縮されたデータがPPからBPへ出てきて、次に右隣となる圧縮データ15を書き込む領域まで圧縮されたデータが流れている。丁度、画面には圧縮器2と圧縮データ15が書き込まれる領域間となるBPの経路上に一連の圧縮されたデータの一部となるデータの'c'（これも16進数で表示）が示してある。

【0041】これらPPの回路から読み出される構成情報と、圧縮器により圧縮された後のデータの流れについて、両者のデータ量と速度の比較をする。前者（回路から読み出される構成情報）のデータ量は正に1PCAcel1分（4bit×1024words）であり、後者（圧縮された後のデータ）は圧縮されているので、そのデータ量は1PCAcel1分に満たない。このため、前者は後者に比べてデータ量が多い。そして、両者は、圧縮処理の開始と終了の間に異なるデータ量の入力と出力として同じ圧縮器で扱われる所以、速度に関して前者は後者よりも高速である。以上で、PCA上で簡単な圧縮の実現の事例を説明した。このような圧縮器をさらに効率的な圧縮能力のある回路を採用して発展的に応用することで、図1の手法適用や図3のような第1の実施形態（外部退避）を可能にできる。

【0042】発展的な応用例を具体的に述べると、PCAはBPに設定する経路を通じて外部端子ともデータの入出力を行う。従って、このPCAでの説明では、圧縮（符号）器の出力が、この圧縮（符号）器の隣にあるPPの領域をメモリとして格納されている。このメモリに当たる箇所を、単に外部端子に置き換えて、PCAのBPに設定する経路を通じて外部端子からデータを入力することで、図3に示すような第1の実施形態（外部退避）が実現されるのである。つまり、単純に圧縮（符号）器をPCAの外部端子に接続させて設けたり、あるいは圧縮（符号）器の出力からPCAの外部端子までの経路を長く伸ばすなどしてBPに設定したりすれば良いのである。

【0043】（第2、第3、第4の実施形態に関する実施例2）図12は、本発明の実施例2を示すPCA上の伸張前の準備の画面および説明図である。圧縮したデータから回路1と同じ回路を復元する伸張について述べる前に、圧縮データを格納している領域が1PCAcel1に満たないため、正に1PCAcel1にして伸張の段階で扱い易いように残る領域に無意味な書き込みを行う。また、この圧縮データを格納する1PCAcel1で残りの領域へは既にBP上に圧縮器からの経路が設定されたままになっている。そこで、一旦圧縮器のBPとPPとの入出力に当たる位置に通過器18を置く。この通過器18は、入力されたデータを何も変化させずに

そのまま出力する回路である。

【0044】図12のデータ補充動作では、外部から無意味なデータを注入17し、通過器18を介して圧縮データの格納している1PCAcel1の残る領域へ補充している様子19を示している。図12に示すPCAのBPのシミュレータ画面には、左端の最上段に位置する外部入力から通過器18までの間に、BPの経路が設定され、その経路上に無意味なデータとなる一端の'6'（16進数表示されたデータ）が外部から注入されている。また、通過器18から圧縮データを格納する領域へのデータの補充については、通過器18からその圧縮データを格納している領域までの間のBPの経路上に、通過器18を介して送られているデータ、つまり圧縮データの格納で残る領域を補充するデータ19が確認できる。このデータ19は、図12では通過器18と圧縮データの格納する領域とのBP上の経路にある'7'（これも16進数表示）である。

【0045】このように、通過器18を介して外部から無意味なデータ「コード」を任意の量（領域の残りの分のデータ列の長さ）を注入することにより、この後に説明する伸張のための圧縮データの活用が（1PCAcel1単位の読み出し操作で済ませることで可能となり）簡便にできる。なお、ここで説明した伸張前の準備は、シミュレータ画面上にブロックで示される領域、つまり1PCAcel1単位の部分的な書き込みと読み出しが実現できれば、必要はない。さらに、圧縮器と伸張器にデータ列の終わりを識別して、圧縮動作にはデータ列の終了後には経路消去と、伸張側で圧縮されたデータが書き込まれていない領域を読むとスキップする機能を付加することでも、このような準備のための操作を行わなくても済む。

【0046】図13は、本発明の実施例2を示す圧縮データと伸張器の画面および説明図である。図13には、圧縮したデータから、元の回路を復元する伸張、つまり第2の実施例をPCAに適用した場合の説明に移る。まず、PCAのPPに伸張器4を設ける。図13のPCAのPPのシミュレータ画面は左上隅から順に回路X1、この下側に接して通過器18、この通過器18は以前圧縮器のあった領域の一部（左上1/4の範囲）に上書きしてプログラムされたものである。また、この右隣に圧縮データを格納した領域があり、この領域は外部から無意味なデータを補充して丁度1つのブロック（1PCAcel1）分としている。そのまた右隣のPPに伸張器4を配置する。

【0047】ここで用いる伸張器4は、PCAのPPで4ブロック（2×2PCAcel1s）の領域を占めており、圧縮器と同じサイズである。ただし、圧縮器と伸張器の内部の回路構造を細かく確認すると、圧縮器よりも伸張器が比較的小スペースであり、この伸張器の回路設計へ更に手を加えれば、圧縮器以上にコンパクトにで

きる。ちなみに、原理的な裏付けからも、一般的に圧縮器より伸張器の構成は単純になると言われる。図13に示すPCAのPPであるシミュレータ画面の様子では、伸張器4の上側に接して点線の四角の枠を描いている。これが伸張後の回路Yが実現される領域となり、その領域の目印に四角を置いている。

【0048】図14は、伸張動作の概略を示す画面および説明図である。図14の伸張動作の概略は、前に説明した図13のPCAのPPの状況を概略的に示している。また、図14は、伸張動作を開始して、その伸張動作の途中の段階である。従って、伸張データ42が既にPCAのPPで回路Yの一部となり現れている。その部分が伸張器4の上側に接する領域である。

【0049】図15は、伸張データの流れを示す画面および説明図である。伸張動作の同時刻におけるPCAのBPでの状況を説明する。図15の伸張データの流れが、PCAのBPでその伸張動作をシミュレーションしている画面である。PPに実現されている圧縮・伸張に関する回路と、データを書き込んだ場所などに対応するBPの位置・領域を、点線・破線・実線の四角で描いて目印に示している。図15のシミュレータ画面で、ほぼ真ん中に描かれている実線の小さな四角は、圧縮データ15（と補充データ）をPPに書き込んである領域に対応したBPである。この場所でPPからBPはそのデータを読み出し、右隣の伸張器4の入力位置に対応するBPまで経路を設定してある。この経路に沿って圧縮データから読み出されたデータが伸張器のある側のBPへ運ばれている様子が見て取れる。つまり、この図15では、「6（16進数表示のデータ）」が経路上にある。

【0050】また、図15に示すPCAのBPでのシミュレータ画面において、最も右側に位置する実線で描かれた大きな四角が、PPに伸張器4が実現されている所に対応するBPである。この四角の左上がPPの伸張器4の入力と出力が位置するBPである。この位置から上に接した場所へ伸張データ42が送られるために設定された経路がBPに確認できる。これら圧縮データ15から読み出されたデータと、伸張器4により伸張（復元）された後のデータの流れ16について、両者のデータ量と速度の比較をする。前者（圧縮データ15から読み出されたデータ）のデータ量は、後者（伸張された後のデータ42）のデータ量（復元された回路の構成情報に当たるので丁度1PCAcell1分-4bit×1024words）に比べてデータ量は少ない。ここで、圧縮データが1PCAcell1分のデータ量を満たさないために、この伸張処理を前にして圧縮データに加えられた補充データは、伸張に対して無意味である。そして、両者は、伸張処理の開始と終了の間に異なるデータ量の入力と出力として同じ伸張器4で扱われる所以、速度に関して前者は後者よりも低速である。

【0051】なお、図15で、左端に接している破線で四角を描いたブロックで、黒丸の点「.」が中央に印されている所は、通過器18がPPに存在するBPの領域を示している。この場所のBP、つまり通過器4の伸張前の準備操作として利用し終えた後のこの場所には、PPとBPの間の接続を切り離していない（この位置のPPの回路、通過器4を停止する命令をわざわざ発行していないので、回路が動作している状態にある）ために、このブロック中央に黒丸の点「.」の印が残ったままとなっている。また、点線で描かれた四角の領域は、先の圧縮に関係した回路のPPに対応したBPを印しているものである。

【0052】図16は、圧縮・伸張で回路復元を行う画面および説明図である。最後に、圧縮されたデータを伸張することで、回路Yが生成されたPCAのPPの状況を、図16の圧縮・伸張で回路の復元を行う。図16は、前述の図14と同じで、シミュレータ画面のほぼ中央の1ブロック（1PCAcell11）分の領域が圧縮データ（と補充データ）を書き込んだ所である。そして、画面の最も新側に位置する4ブロック（2×2PCAcells）分の範囲が伸張器4である。さらに、伸張器4の上に接している1ブロック（1PCAcell11）分の領域が、最終的に圧縮データを伸張したデータ41を書き込んだ場所であり、圧縮データを復元して構成した回路Yになっている。また、図16に示すPCAのPPにおいて、左上隅に当る箇所が圧縮する前の元回路X1である。この元回路X1と、圧縮データを復元した回路Y（元回路X1から右へ3ブロック、下へ1ブロック移動した箇所に該当する）とを比べると、回路内部の構造が同じであることが、図16のシミュレータ画面により確認できる。

【0053】以上で、RHとしてPCAの場合で、回路の構成情報を圧縮し（第1の実施形態をPCAに適用）、また、その圧縮したデータを伸張して（第2の実施形態をPCAに適用）、元の回路と同じ構造の回路Yを生成した。しかも、全てPCAの上でこれらの伸張動作が実現できること（圧縮と伸張それを組合せた第3の実施形態をPCAに適用）を示した。ここで説明した圧縮と伸張は単純なものであるが、この伸張動作のみや、これらの圧縮と伸張の両方の動作を併せて発展的に応用することで、先に挙げた図2に示す手法や、図4、図5、図6をそれぞれに用いて説明した第2の実施形態（外部プログラム）、第3の実施形態（異なる回路伸張）、第4の実施形態（回路圧縮・伸張）を実現することができる。

【0054】発展的な応用について具体的な事項としては、上述の説明で元回路X1と回路YのPCAのPPの位置を一致させた場合が、図6に示す第4の実施形態（回路圧縮・伸張）である。PCAのPP上で元回路X1と回路Yの位置を一致させるためには、圧縮器や伸張

器の配置や入出力データの経路を変更することで、容易に実現できる。ただし、回路圧縮・伸張では、図6に示すように、元回路Aと入力回路Cの位置をわざわざ一致させる必要性が必ずしもあるのでは無く、前記PCAでの圧縮・伸張の説明のように位置が異なっていても構わないケースもある。

【0055】また、図5に示す第3の実施形態（異なる回路伸張）は、複数の圧縮データ（Da, Db）がある点が、今回の説明と異なる。しかし、これらの圧縮データは複数の回路から、それらの構成情報を圧縮したものに過ぎず、その圧縮の処理は今回のPCA上の圧縮と同じである。ただ、この圧縮データを予め外部で同じ圧縮アルゴリズムのプログラムを用いて準備することも有可能。いずれにせよ、圧縮データを複数準備しておけば、その後の伸張は全く同じである。

【0056】さらに、図4に示す第2の実施形態（外部プログラム）は、このPCA上の伸張動作で圧縮データをBPの経路に沿って伸張器へ流す箇所を、外部端子からBPに経路を設定することに代えて実現できる。実施例1におけるPCA上の圧縮で、図3のような第1の実施形態（外部退避）への応用に関して、BPにより外部端子への出力データが扱えると説明した。図4に示す第2の実施形態（外部プログラム）は、正に図3のような第1の実施形態（外部退避）とは逆の形態に該当する。つまり、第2の実施形態（外部プログラム）では、BPにより外部端子からの入力データを扱うのである。単純にPCAの外部端子に隣接させて伸張（復号）器を設けたり、あるいは、PCAの外部端子から伸張（復号器の入力までの経路を長く伸ばしたりするなど）して、BPに設定する。外部端子からBPへデータが入力した後、伸張器のデータ処理は全く違いがなく、第2の実施形態は容易に実現できる。

【0057】実施例1（PCA上の圧縮）の最後で述べたように、圧縮器だけでなく伸張器も併せて、さらに効率的な圧縮・伸張能力のある回路を採用して応用する（拡張した幅のデータに対応させたり、圧縮と伸張のアルゴリズム自体より良いものを選択して適用したりする）ことも検討する範囲である。

【0058】（従来例3との相違点）なお、本発明における従来例3との相違点について述べる。従来例3では、論理回路を実現するプログラム論理回路部と回路情報入力制御部とが別々の機構であるのに対して、本発明では、上記のような区別を設けないので、任意の回路を実現する所と回路の圧縮や伸張する所を自由に設定できる点で全く異なる。また、従来例3では、回路情報入力制御部に圧縮する機構を含んでいるのに対して、本発明では、圧縮や伸張の機構そのものもプログラムして実現する点で、全く異なっている。このため、圧縮や伸張する回路を任意の位置に移したり、不要な時には消去（別の機能を上書き）したり、自在に扱うことができる。特

に、圧縮のためのアルゴリズムについても、用途に合わせて、その都度、プログラム変更し対応することができる。

【0059】また、従来例3では、回路記憶部という専用の格納エリアを持っているのに対して、本発明では、回路記憶部という格納の専用エリアを設けず、任意の位置に一時的にでも回路構成情報やその圧縮データまたその他のデータも格納することができる点で、全く異なっている。また、従来例3では、回路情報を外部から回路情報入力制御部に入力し、ここで入力された情報のコマンド（3種類）を識別して、それに従って圧縮したり、あるいはしないで、プログラマブル論理回路部内のコンフィグレーションメモリに格納している。これに対して、本発明では、回路情報を入力してRH領域内に生成された回路に格納され、その回路からのデータがRH領域内の圧縮器で変換されて、圧縮データになり、RH内に生成されたメモリに格納される。従って、従来例3では、必ずリピートフレーム（圧縮形）ノーマルフレーム（伸張形）、リファレンスフレーム（通常形）の3種類のデータを識別できるコマンドとともにに入力されるのに対して、本発明では、種類はなく、コマンドも不要であり、回路情報のみを入力すればよい。

【0060】また、従来例3では、全てハードウェアで構成され、回路情報入力制御部、回路情報編集部、コンフィグレーションメモリと回路素子を含むプログラマブル論理回路部とを設ける必要があるので、本発明では、回路と、圧縮器と、伸張器と、メモリが、いずれもソフトウェアで必要に応じて任意に生成できる。さらに、従来例3では、データを入力してプログラマブル論理回路装置で回路を再構成し、データを出力しないのに対して、本発明では、圧縮器で変換された圧縮データを外部に退避させることができる。すなわち、外部への出力機能を有している。そして、従来例3の発明の目的は、プログラマブル論理回路装置の再構成時間を短縮することであるのに対して、本発明の目的は、入出力データの容量抑制と、高速な入出力動作を可能にして、圧縮器と伸張器を任意に実現し、ハードウェアリソースの有効活用を図ることであり、両者は目的が異なっている。

【0061】

【発明の効果】以上説明したように、本発明によれば、入出力データには圧縮（符号化）したデータをデータ処理回路の内部で伸張（復号化）すること、と圧縮（符号化）も内部ですること、そしてこれら組み合わせ（第4の実施形態）と、様々な形態が適用できる。その結果、入出力データの容量抑制、高速な入出力動作をするデータ処理回路の実現が可能になること、および、これを再構成可能なハードウェアで回路の構成情報を当てはめると、回路の構成情報を読み出しや書き込み処理をこれまでよりも小容量のデータで素早く実行できること、などの顕著な効果を奏する。

【0062】また、発展的な適用として、特にPCAのように自律的かつ動的な再構成可能なハードウェアRH内部での圧縮・伸張器は、それ自体も再構成可能な回路として必要な時、必要な位置に自在に実現できるので、より柔軟な応用が可能となる。また、圧縮した回路の構成情報は、回路それ自体よりも小規模な領域に格納でき、回路が動作する必要な時だけ伸張して復元されればよい。先に使用した圧縮データを消去さえしなければ、回路が停止または待機する時には、特に改めて回路の構成情報を圧縮させる手順をせずに済む。

【0063】さらに、本発明の適用は、単にRHの容量（領域）の有効な活用という利点に止まらない。これまで動作に係らず必要とされたデータを回路に入出力する経路を、同じ領域に異なる回路を順に復元する場合、そのまま再利用することも可能となる。つまり、この経路に必要なハードウェア・リソースが削減され、今まで回路設計と同時に必要とされた入出力経路の設計も不要にできる。

#### 【図面の簡単な説明】

【図1】本発明の第1の実施形態を示す回路構成情報の圧縮の説明図である。

【図2】本発明の第2の実施形態を示す回路構成情報の伸張の説明図である。

#### 【図3】本発明の第1の実施形態を示す圧縮の利用例（外部退避）の説明図である。

【図4】本発明の第2の実施形態を示す伸張の利用例（プログラム）の説明図である。

【図5】本発明の第3の実施形態を示す異なる回路伸張の利用例の説明図である。

【図6】本発明の第4の実施形態を示す回路圧縮・伸張の利用例の説明図である。

#### 【図7】従来におけるRH利用例1（プログラム）の説

明図である。

【図8】従来におけるRHの利用例2（複製機能）の説明図である。

【図9】本発明の実施例1を示す圧縮器と動作の状況の画面およびその説明図である。

【図10】本発明における圧縮の動作の概略原理を示す画面およびその説明図である。

【図11】本発明における圧縮データの流れを示す画面およびその説明図である。

【図12】本発明の実施例2を示すデータの補充動作の画面およびその説明図である。

【図13】本発明の実施例2を示す圧縮データと伸張器の画面およびその説明図である。

【図14】本発明における伸張動作の概略を示す画面およびその説明図である。

【図15】本発明における伸張データの流れを示す画面およびその説明図である。

【図16】本発明における圧縮・伸張で回路復元を行う画面およびその説明図である。

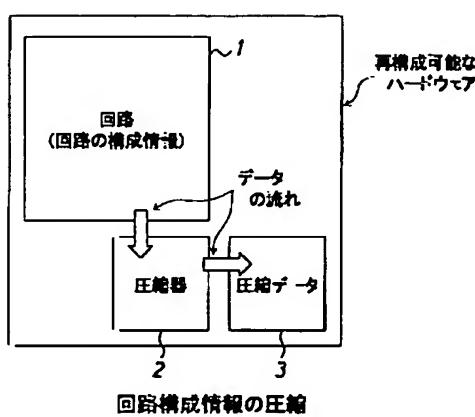
【図17】本発明の実施例1を示すPCAに実現した連長圧縮器の画面図である。

【図18】本発明の実施例2を示すPCAに実現した連長伸張器の画面図である。

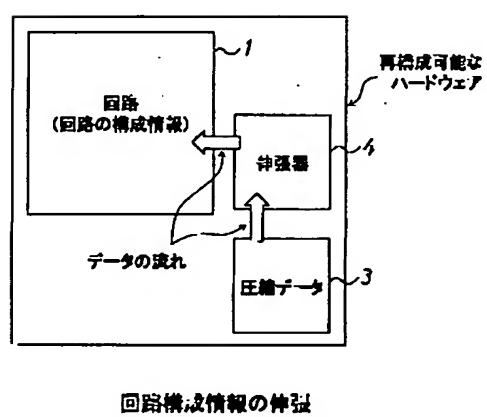
#### 【符号の説明】

1…回路（回路の構成情報）、2…圧縮器、3…圧縮データ、4…伸張器、5…回路A／回路B、回路A／回路C、7, 8…圧縮データ（D a）（D b）、9…回路A'／回路B'、10…回路C'、11…回路D'、15…圧縮データ、16…データの流れ、17…外部から補充データの注入、41…伸張データ（回路Y）、42…伸張データ（回路Yの一部）。

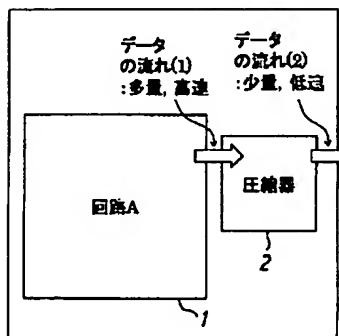
【図1】



【図2】

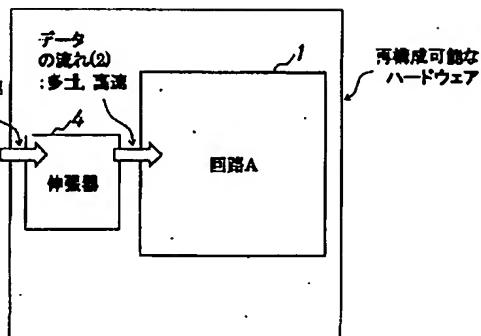


【図3】



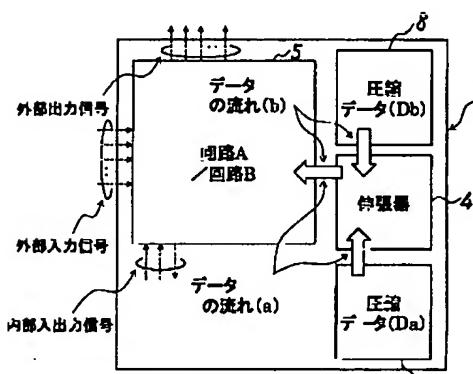
圧縮の利用例(外部退避)

【図4】



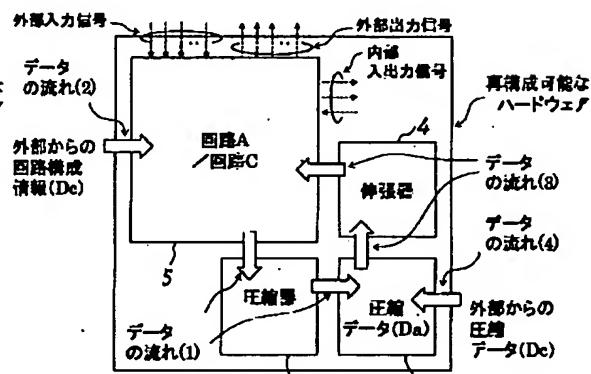
伸張の利用例(ブロック図)

【図5】



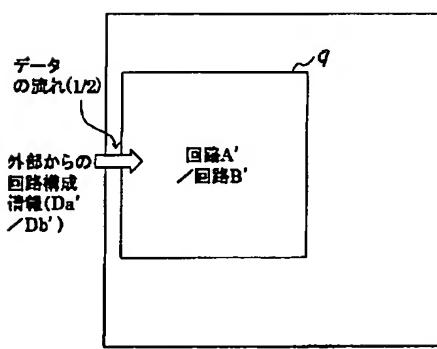
異なる回路伸張の利用例

【図6】



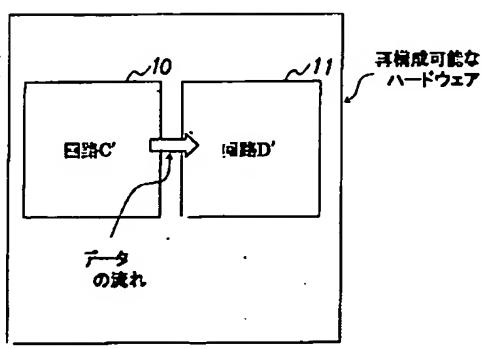
回路圧縮・伸張の利用例

【図7】



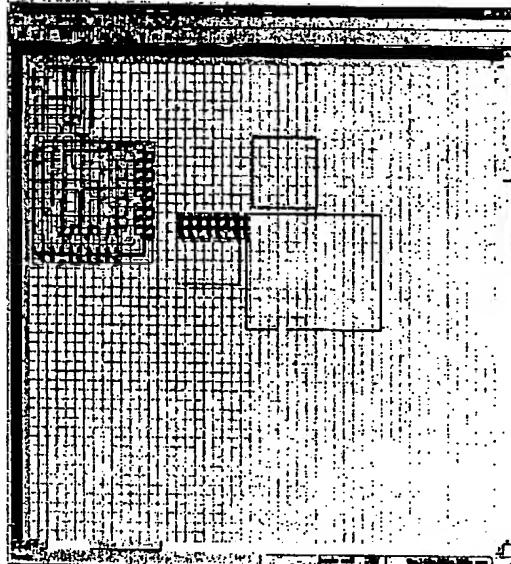
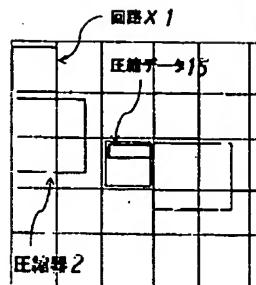
RIIの従来例1 (プログラム)

【図8】



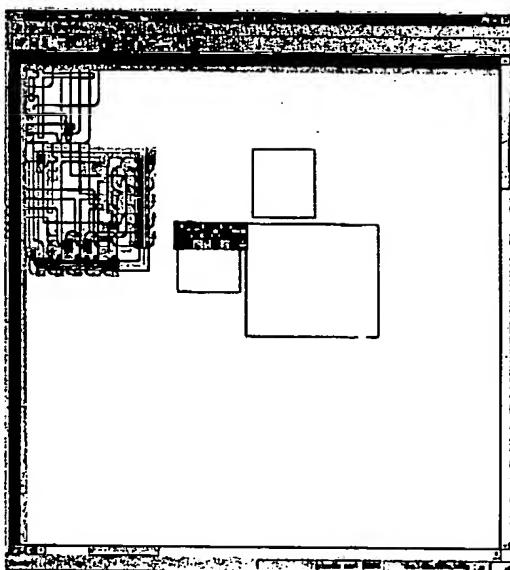
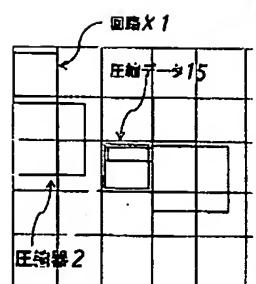
RHの従来例2 (複製機能)

【図9】



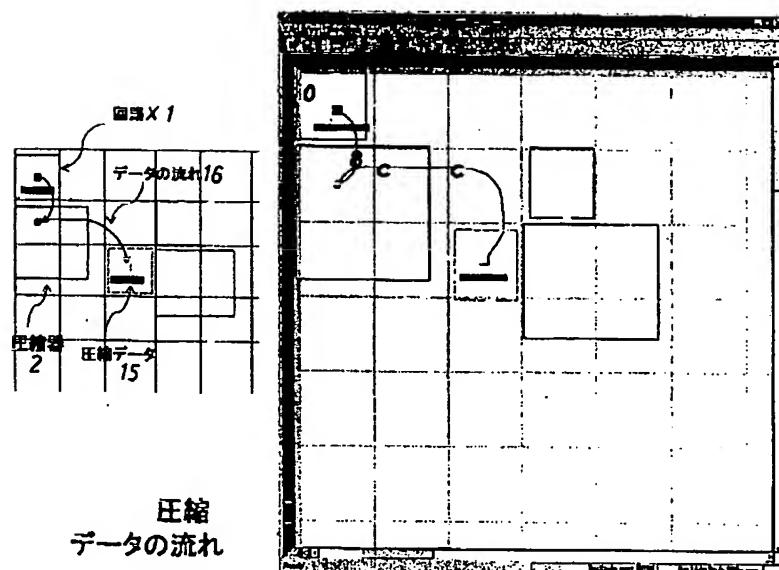
圧縮器と  
動作の状況

【図10】

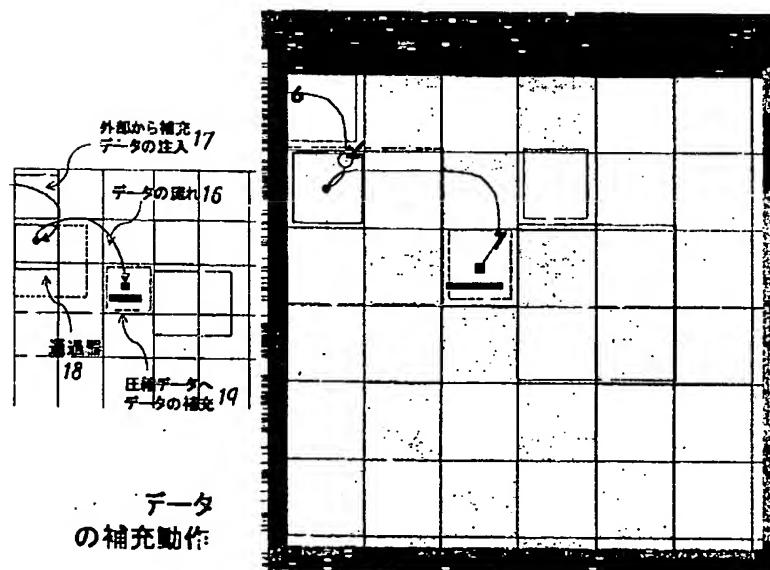


圧縮  
動作の概略

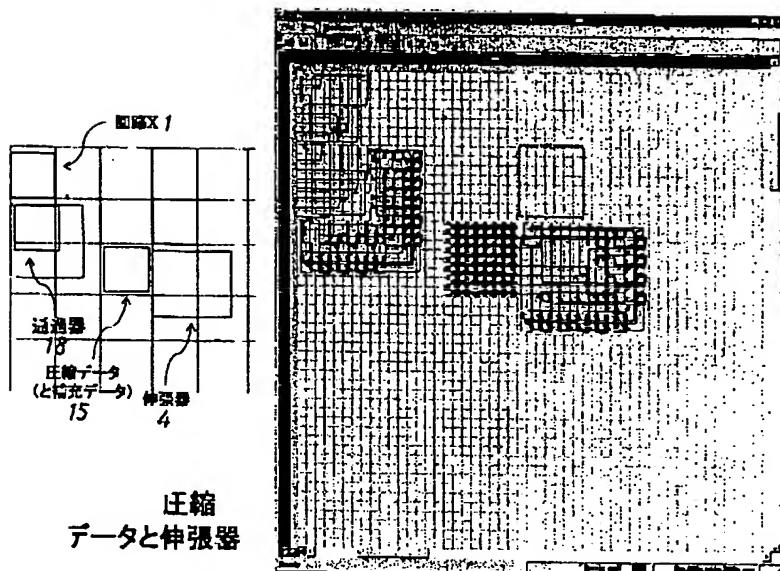
【図11】



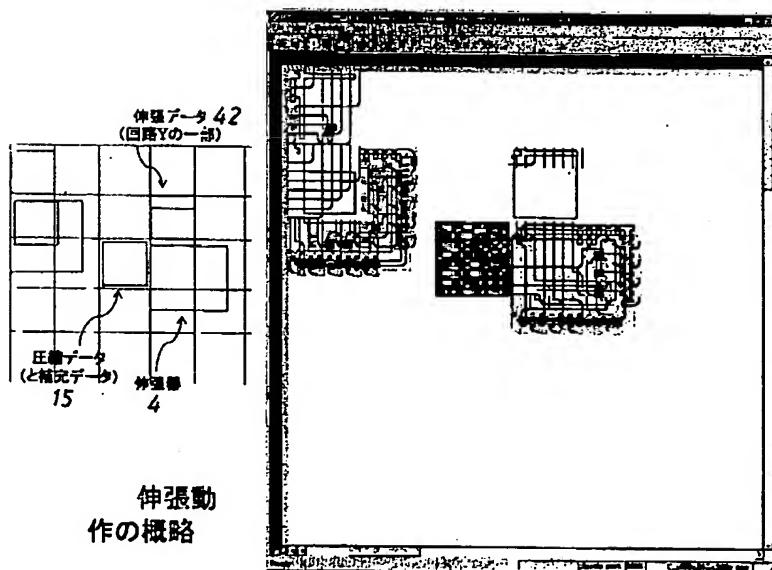
【図12】



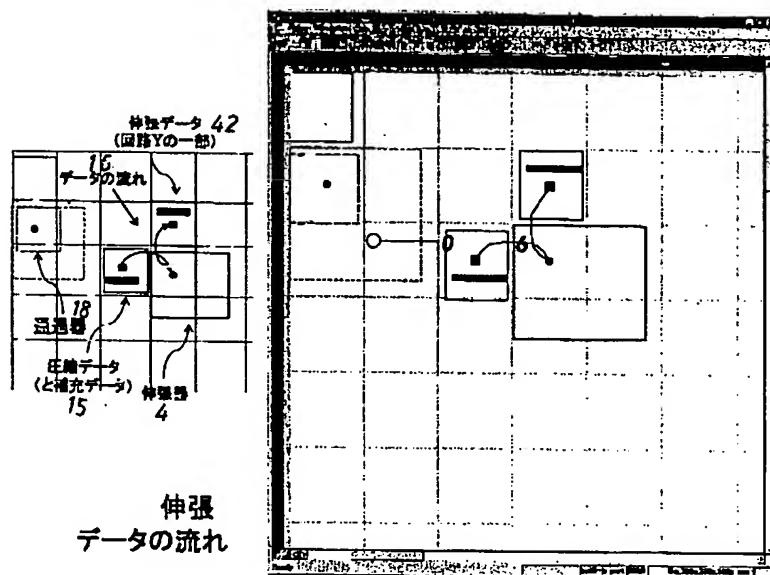
【図13】



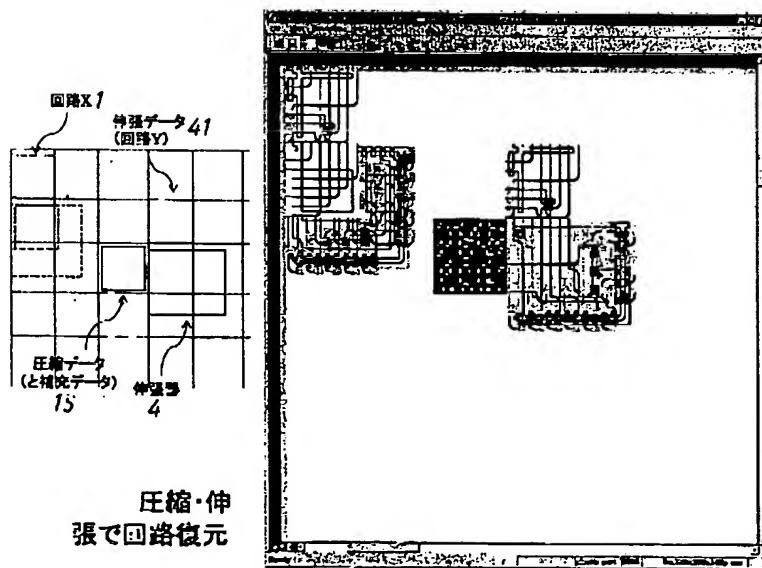
【図14】



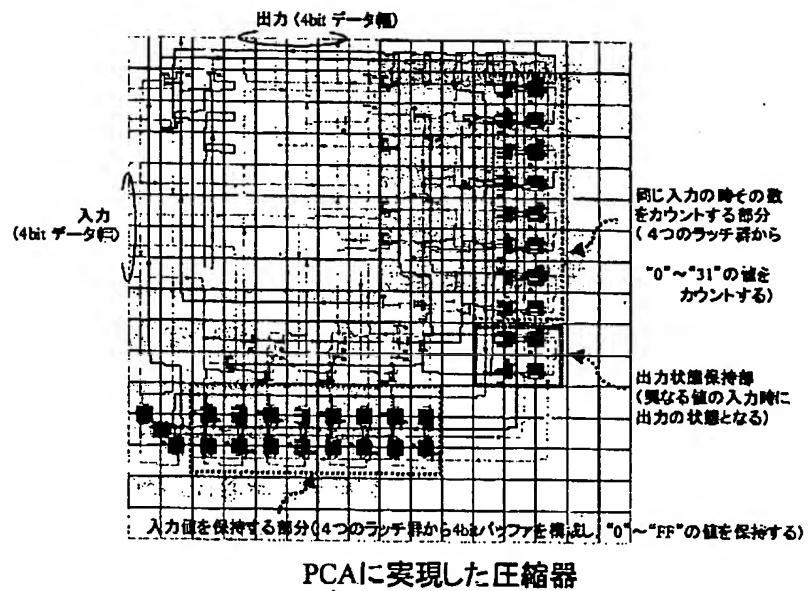
【図15】



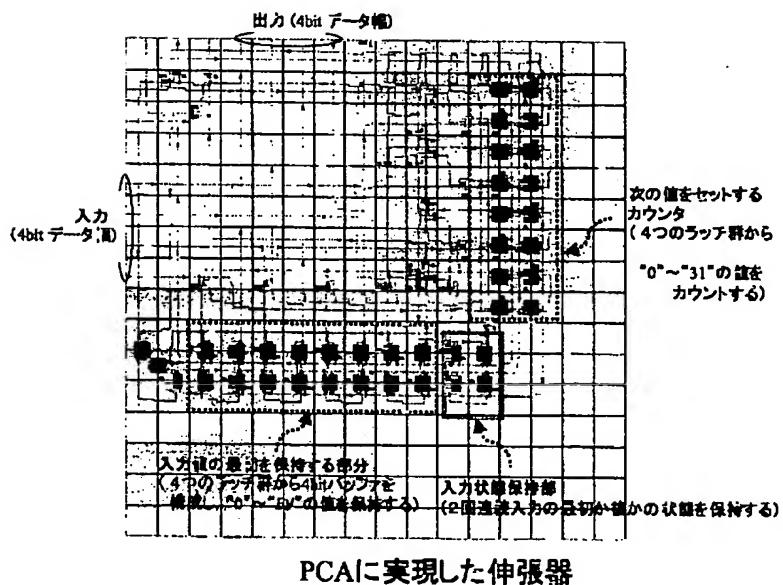
【図16】



【図17】



【図18】



フロントページの続き

(72)発明者 稲森 稔  
東京都千代田区大手町二丁目3番1号 日  
本電信電話株式会社内

(72)発明者 中根 良樹  
東京都千代田区大手町二丁目3番1号 日  
本電信電話株式会社内

F ターム(参考) 5J042 BA01 BA08 BA11 DA01 DA04